UDACITY

< Return to Classroom

DISCUSS ON STUDENT HUB

Machine Translation

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

CONGRATULATIONS

Congratulations on completing this project. You've demonstrated strong understanding and mastery of the concepts. Well done.
for general reading on NMT, here are a few helpful materials on machine translation to further grow in the field.

A Machine Translation tutorial from NVIDIA:

- Part 1: https://devblogs.nvidia.com/introduction-neural-machine-translation-with-gpus/
- Part 2: https://devblogs.nvidia.com/introduction-neural-machine-translation-gpus-part-2/
- Part 3: https://devblogs.nvidia.com/introduction-neural-machine-translation-gpus-part-3/
  Another cool tutorial I found: https://medium.com/@ageitgey/machine-learning-is-fun-part-5-language-translation-with-deeplearning-and-the-magic-of-sequences-2ace0acca0aa

Hope you enjoy these readings. Please be safe out there and keep up with the good work!

Submitted Files

The following files have been submitted: `helper.py`, `machine_translation.ipynb`, `machine_translation.html`

All required files were submitted.

Thank you for submitting all the required files

Preprocess

The function `tokenize` returns tokenized input and the tokenized class.

`tokenize` is flawlessly implemented.

Well done implementing this correctly

The function `pad` returns padded input to the correct length.

`pad` nicely done

Nicely done. Good job validating the input of the length, we should never assume that the user would enter a valid value

```
if length is None:
        length = max(len(s) for s in x)
```

Models

The function `simple_model` builds a basic RNN model.

`simple_model` nicely done

This model is simple and doesn't really learn a lot of representations. To improve the accuracy you can increase

the number of units. However it may start overfitting, which could be fixed using dropout.

The function `embed_model` builds a RNN model using word embedding.

`embed_model` nicely done

Well done with this model. The embedding layer adds a lot more context for the model to learn from. It is normal that the accuracy increases greatly compared to the previous model.

The Embedding RNN is trained on the dataset. A prediction using the model on the training dataset is printed in the notebook.

The function `bd_model` builds a bidirectional RNN model.

`bd_model` nicely done

Nicely done. We are now learning representations by traversing the sequence from both directions. It is normal that the accuracy decreases compared to the previous model. In the next models, we will see the power of these layers when combined together.

The Bidirectional RNN is trained on the dataset. A prediction using the model on the training dataset is printed in the notebook.

The function `model_final` builds and trains a model that incorporates embedding, and bidirectional RNN using the dataset.

100% on implementing `model_final`

Amazing accuracy. You have done a great job with this model, building the final model using all the learning and building blocks from the previous models.

Prediction

The final model correctly predicts both sentences.

Spot on translation

Epoch 30/30
110288/110288 [==============================] - 27s 243us/step - loss: 0.0784 - acc: 0.9775 - val_loss: 0.0925 - val_acc: 0.9733
Sample 1:
il a vu un vieux camion jaune <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
Il a vu un vieux camion jaune
Sample 2:
new jersey est parfois calme pendant l' automne et il est neigeux en avril <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
new jersey est parfois calme pendant l' automne et il est neigeux en avril <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>

DOWNLOAD PROJECT

RETURN TO PATH

Rate this project